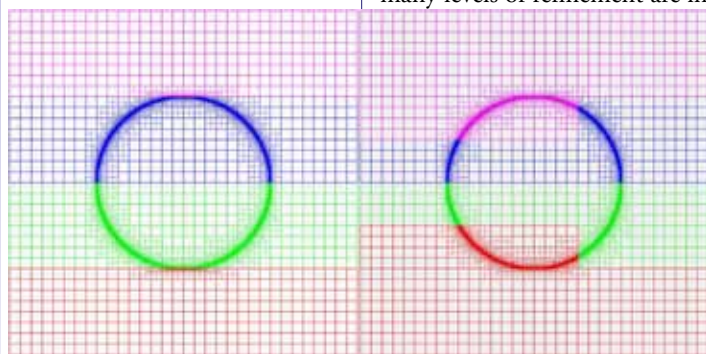


# Development of a New Adaptive Mesh Refinement Framework for Multiphysics Simulations

William W. Dai, HPC-5

**Fig. 1.** The distribution of cells among four processors in a mesh with ten levels before (left image) and after (right image) the load balance. Each color represents the cells for which one PE is responsible. Before the balance, two middle processors each have about 0.6 million cells, whereas the other two processors each have about 6000 cells. After the balance, the middle two processors each have about 305,000 cells, and other two processors each have about 294,000 cells.



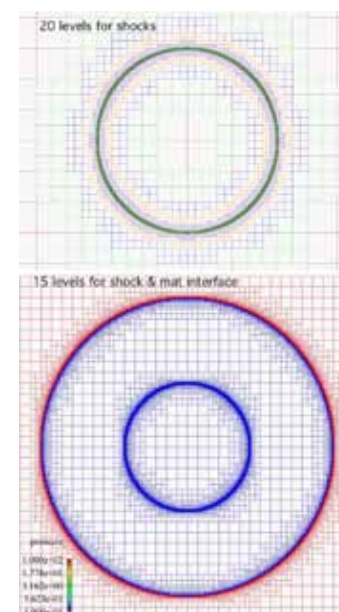
**A**daptive mesh refinement (AMR) in analytical computation is the selective development of grids within grids where more detail is required than available using the parameters of the standard grid coordinates. There are three kinds of AMR. (1) Block-based AMR refines a predefined block when any cell within the block is marked to be refined. The advantages of the block-based AMR include the nature of structured meshes of each block and relatively simple data structures. But it sometimes over-refines meshes. (2) Cell-based AMR [1] refines only those cells that are supposed to be refined. But in cell-based AMR, cells, including the cells that are not refined, are typically treated cell by cell, even for structured meshes. Therefore cell-based AMR often loses the advantage of the nature of structured meshes. (3) Patch-based AMR dynamically groups to-be-refined cells to form rectangular patches and thus it combines the advantages of block- and cell-based AMR, i.e., the nature of structured meshes and the sharp regions of refinement. But patch-based AMR has its own difficulties. For example, existing patch-based AMR typically cannot preserve symmetries of physics problems. Since patches in patch-based AMR are dynamically generated and have different sizes, efficiently managing patches and communications in parallel environments presents a challenge when many levels of refinement are involved.

In this work, we have developed a patch-based AMR framework through which refined meshes are able to preserve symmetries of physics problems. Through the framework, the grid cells are refined near shock fronts, shear layers, and material interfaces. The development of the framework has been focused on its use

for multiphysics simulations and compressed material data structures. The approach for AMR consists of clustering, symmetry preserving, mesh continuity, flux correction, communications, management of patches, and dynamic load balance. The special features of this patch-based AMR include symmetry preservation, efficiency of refinement across shock fronts and material interfaces, special implementation of flux correction, and patch management in parallel computing environments.

For symmetry, if to-be-refined cells are symmetrical with respect to the x- or y- or z-axis, we want the refined cells determined through clustering algorithms to preserve the symmetry. We have developed additional steps, on top of a k-means algorithm developed by MacQueen [2], to preserve the symmetries. We have also implemented a procedure to ensure that the resolution in the resulting mesh through multiple levels of refinement smoothly changes from location to location. If a grid cell undergoes  $m$  levels of refinement, its immediate neighboring cells must undergo at least  $(m-1)$  levels of refinement.

One of the advantages of patch-based AMR is that physics solvers are implemented on uniformly spaced structured meshes on each patch. To do that, the values of physics variables on the ghost cells of patches have to be obtained before physics solvers can be executed. We have implemented a procedure in which each computer processor communicates with each of its neighboring processors only once for



**Fig. 2.** The top mesh has 20 levels of cells near a circular shock front, and the bottom mesh has 16 levels of cell near a shock front (the outer circle) and a material interface (the inner circle). The cells of each level in the top image are shown through one color.

the values of ghost cells during one time step, no matter how many levels of refinement or which physics solvers are involved. Typically, the thickness of the layer of ghost cells depends on the physics solvers to be used, and the higher-order a solver is, the thicker the layer of ghost cells. In our framework, the thickness of the layer is a parameter, which can be set to the one appropriate for the physics solvers.

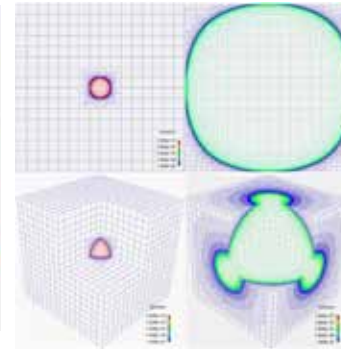
The number of patches in patch-based AMR may dramatically increase with the size of physics problems and the number of levels of refinement. The management of these patches may significantly influence the performance of AMR in parallel environments. We have developed an approach in which each processor manages only the minimum number of patches needed for updating physics variables on the processor. This approach yields far superior performance.

In a parallel computing environment, each computer processor is responsible for a certain amount of workload. Because of the dynamic nature of shock fronts and material interfaces, the workload of each processor also dynamically changes. We have implemented an approach for dynamic load balance. The approach is not for redistribution of workload but for the minimum data movement based on the existing workload and the connectivity of the subdomain each processor is responsible for. Figure 1 shows the distributions of grid cells in a mesh of 10 levels among four processors before and after the load balance. In the figure each color is the region for which one processor is responsible. Before the load balance, two middle processors each have about 0.6 million cells, whereas two other processors each have about 6000 cells. After the load balance, the middle two processors each have about 305,000 cells, and of the other two processors have about 294,000 cells.

**Numerical Examples.** Typically we refine shock fronts, shear layers, and material interfaces. For shocks and shear layers we use shock jump conditions and derivatives as the criterion for refinement. For material interfaces, we refine any cell with more than one material or mixed cell, and any pure cell whose immediately neighboring cells contain other materials. Figure 2 shows two meshes, the top one with 20 levels of cells near a circular shock front, and the bottom one with 16 levels of cells near a shock front (the outer circle) and



**Figure 3.** One eighth of a mesh with 10 levels of cells near a spherical shock front. The cells at each level are supposed to be buried in the cells of the previous level.



**Figure 4.** Distributions of pressure in 2D and 3D simulations at the initial time (the left images) and when the circular and spherical shocks reach the boundaries of the simulation domains. The refinement follows shocks propagating outward.

a material interface (the inner circle). Each color in the top image represents one level of cells. Only the first few levels of coarse cells are recognizable in the figure. The numbers of the 20 levels of cells in the top mesh, from the coarsest to finest, are 256, 304, 592, 1166, 2320, 4624, 9232, 18448, 36880, 73744, 147440, 294864, 589520, 1178128, 2352432, 4690128, 9321936, 18406656, 35890816, and 68047520, respectively. If one cell of a base patch were fully refined 19 times, the number of the finest cells in the cell would be 274,877,906,944.

The framework also works for the 3D case. Figure 3 shows one-eighth of refined cells near a spherical shock front with 10 levels of cells. Cells at each level are actually buried in the cells of the previous level. The white dots in the last image in the figure indicate that the cells at the second-finest level do not undergo the refinement at these spots.

This AMR framework has been applied to some simple hydrodynamics calculations. Figure 4 shows the distributions of pressure in 2D and 3D simulations at the initial time (the left images), and when the circular and spherical shocks reach the boundaries of the simulation domain. The refinement follows the shock propagating outward.

We will also work on physics solvers involving multimaterials and compressed material data structures. When tens of materials are involved in large-scale simulations, compressed material data structures become very important.

**For more information contact William W. Dai at [dai@lanl.gov](mailto:dai@lanl.gov).**

[1] M. Gittings et al., *Comput. Sci. Disc.* **1**, (2008); 015005, doi:10.1088/1749-4699/1/1/015005.

[2] J.B. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proc. 5th Symp. Math. Stat. Probab.*, University of California Press (1967).

#### Funding Acknowledgments

DOE, NNSA Advanced Simulation and Computing Program Radiation Hydrodynamics on Roadrunner Project and Productivity Project